# Terraform Task

**Name:Shaik Khaja Basha**
**Batch : Batch 11**
**Date : 24.07.2025**
**Task : ignore canges & replace triggred**

1. **Explore the life cycle of ignore_changes and replace_triggered_by** ?

**Ans:** In Terraform, ignore_changes and replace_triggered_by are lifecycle meta-arguments that control how resources are managed. ignore_changes prevents Terraform from attempting to update specific attributes that are being modified outside of Terraform, while replace_triggered_by forces a resource to be replaced when specific referenced resources or variables change

**ignore_changes:**
**purpose:**

- To ignore changes to specific attributes of a resource that are being managed outside of Terraform.
- Even if instance_type changes in code or tfvars, Terraform will **not trigger a replacement** or update the resource.

**Use Cases:**

- When an external process (like an Azure Policy) modifies resource attributes, and Terraform should not attempt to revert those changes.
- When dealing with dynamic data that may change after resource creation, but should not trigger a resource replacement.

**Example:**

Ignoring changes to tags on an AWS instance that are being managed by an external tagging process.

**working:**

Terraform will still manage the resource's creation and destruction, but it will not attempt to reconcile the specified attributes with the Terraform configuration.
**Syntax:**

```
        resource "aws_instance" "example" {
                  ami           = "ami-123456"
            instance_type = var.instance_type

  lifecycle {
    ignore_changes = [instance_type]
  }
}
```

**Scenario**:
You manage an Azure App Service. Operations team might change settings manually, and you don't want Terraform to revert their changes every time.

**Example:**

```
lifecycle {
  ignore_changes = [ app_settings ]
}
```

Terraform will not re-apply changes if only app_settings are different in the state vs config.

**Effect:**

- Terraform won't recreate the resource if only the timestamp changes
- Useful when you want to keep the resource stable despite volatile attributes

**replace_triggered_by:**

**Purpose:**
To force a resource to be recreated when specific referenced resources or variables change

**Use Cases:**

- When a resource depends on another resource whose changes should trigger a full replacement of the dependent resource.
- When a resource needs to be replaced when a specific variable or attribute of another resource changes.
- If the local_file.example resource is modified or recreated, null_resource.example will also be replaced.

**Example:**

Replacing a Google Compute Instance when its startup script changes.

**How it works:**

When the referenced resource or variable changes, Terraform will plan a replacement of the resource, effectively destroying and recreating it.

**Syntax:**

```
resource "null_resource" "example" {
  triggers = {
    value = var.value
  }

  lifecycle {
    replace_triggered_by = [
      var.value,
      local_file.example     # or another resource
    ]
  }
}
```

**Example with Random Provider:**

```
resource "random_id" "seed" {
  byte_length = 4
}

resource "null_resource" "dependent" {
  triggers = {
    seed = random_id.seed.hex
  }

  lifecycle {
    replace_triggered_by = [random_id.seed]  # Recreate when seed changes
  }
}
```

**Effect:**

- null_resource.dependent will be destroyed/recreated whenever random_id.seed changes
- Creates an explicit dependency link

Difference between ignore_changes and replace_triggered_by

| **ignore_changes** | **replace_triggered_by** |
|---|---|
| ignore_changes prevents updates to specific attributes | replace_triggered_by forces a full resource replacement. |
| ignore_changes is useful for managing resources where external processes make changes | replace_triggered_by is useful for managing resources that depend on other resources that may change. |
| ignore_changes allows for shared management of a resource between Terraform and external processes | replace_triggered_by ensures consistency by replacing the resource when a specific dependency changes |